Package: pslr (via r-universe)

June 23, 2026

Type Package

Title Public Suffix List Engine

Version 1.0.2

Description A focused implementation of the Public Suffix List (PSL). Bundles a reproducible, pinned PSL snapshot and implements the official prevailing-rule algorithm to answer public-suffix (eTLD) and registrable-domain (eTLD+1) queries. Distinguishes ICANN and PRIVATE rule sections, accepts Unicode and ASCII hostnames via 'punycoder' canonicalization, and supports an explicit, validated offline refresh path. The matcher is compiled with 'cpp11' and requires no external system library. Used as the PSL engine by the 'rurl' package.

License MIT + file LICENSE

Language en-US

URL https://bart-turczynski.github.io/pslr/, https://github.com/bart-turczynski/pslr

BugReports https://github.com/bart-turczynski/pslr/issues

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends R (>= 4.0.0)

Imports punycoder (>= 1.1.0), tools, utils

LinkingTo cpp11

Suggests cucumber (>= 2.0.0), curl, digest, knitr, rmarkdown, testthat (>= 3.0.0), withr

Config/testthat/edition 3

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

Repository https://bart-turczynski.r-universe.dev

Date/Publication 2026-06-23 09:14:18 UTC

RemoteUrl https://github.com/bart-turczynski/pslr

RemoteRef HEAD

RemoteSha 7ea7ff8beeec77216e36fdb41f034819f3533ffd

Contents

`is_public_suffix` *Is a host itself a public suffix?*

Description

`TRUE` exactly when the valid canonical host equals its own public suffix under the selected policy. Returns `NA` whenever `public_suffix()` would return `NA` (missing or invalid input, or an unresolved host under `unknown = "na"`). Under the default `unknown = "default"`, an unlisted single label such as `"madeuptld"` is `TRUE` via the implicit `*` rule; ask `unknown = "na"` to test explicit membership instead.

Usage

```
is_public_suffix(
  domain,
  section = c("all", "icann", "private"),
  unknown = c("default", "na"),
  invalid = c("na", "error")
)
```

Arguments

`domain`	Character vector of DNS hostnames (not URLs). Each element may be a mixed-case ASCII, Unicode, or A-label hostname, a single label, or a hostname with exactly one terminal root dot. See **Input contract**.
`section`	Which rule sections are eligible: `"all"` (default; ICANN and PRIVATE), `"icann"`, or `"private"`. Section filtering happens before prevailing-rule selection, so `"private"` does not silently add ICANN rules; a host matching no rule in the section falls through to the implicit default rule unless `unknown = "na"`.

unknown	"default" (default) applies the spec's implicit * rule, so an unlisted single label is its own public suffix; "na" returns NA when no explicit rule in the selected section matches.
invalid	"na" (default) returns NA for each invalid element without a warning; "error" aborts on the first invalid element, reporting its 1-based index.

Value

A logical vector with length(domain), preserving the names of domain.

Input contract

NA is treated as missing (returns NA), not invalid. Invalid elements include empty or whitespace-only strings, leading or consecutive dots, URL syntax, IPv6 addresses, canonical dotted-decimal IPv4 literals, and labels that fail hostname/IDNA validation. Wrong argument types and non-scalar or unknown option values always abort regardless of invalid.

See Also

public_suffix()

Examples

```
is_public_suffix("com")
is_public_suffix("example.com")
is_public_suffix("madeuptld")
is_public_suffix("madeuptld", unknown = "na")
```

---

psl_refresh *Refresh the cached Public Suffix List from upstream*

Description

Downloads, validates, and publishes a fresh Public Suffix List into the user cache. This is the only function in the package that accesses the network, and only when you call it explicitly.

Usage

```
psl_refresh(
  url = "https://publicsuffix.org/list/public_suffix_list.dat",
  force = FALSE,
  activate = FALSE
)
```

Arguments

`url`	Absolute `https` URL of the list source. Defaults to the official list. URLs with another scheme or embedded credentials are rejected, and a redirect to a non-HTTPS URL is refused.
`force`	When `FALSE` (default), a successfully validated cache younger than 24 hours is reused without a download, respecting upstream download guidance. `TRUE` forces a fresh download.
`activate`	When `TRUE`, the resulting snapshot becomes the active list for the session, exactly as `psl_use()` would activate it. When `FALSE` (default), the cache is updated but the active list is unchanged.

Details

Cache age is measured from the successful network retrieval timestamp; reusing a fresh cache does not advance that timestamp. The download goes to a temporary file in binary mode and must be no larger than a documented maximum (16 MiB). The source is then fully validated – UTF-8, section markers, rule grammar, conflicting rules, and successful canonicalization of every rule – and exact same-section duplicates warn once and are deduplicated. Source and metadata are published only after validation succeeds, using an atomic commit that never exposes a partial or mismatched snapshot. A failed refresh never replaces a valid cache or the active matcher.

Value

Invisibly, a one-row data.frame shaped like `psl_version()` describing the selected cache snapshot, whether or not it was activated.

See Also

`psl_use()`, `psl_version()`

Examples

```
## Not run:
psl_refresh()
psl_refresh(force = TRUE, activate = TRUE)

## End(Not run)
```

---

psl_rules *Rules of the active Public Suffix List*

Description

Returns the explicit rules of the active list as a base data.frame, one row per rule. The implicit default `*` rule is not included.

Usage

```
psl_rules(section = c("all", "icann", "private"))
```

Arguments

section	Which rule sections to return: "all" (default), "icann", or "private".

Value

A base data.frame with columns, in order: rule (original source rule text), canonical_rule (the canonicalized rule, including the *. or ! marker), kind ("normal", "wildcard", or "exception"), section ("icann" or "private"), and labels (integer rule depth, counting a wildcard label). Rows are ordered first by section (ICANN before PRIVATE) and then by source-file order.

See Also

psl_version(), public_suffix_rule()

Examples

```
head(psl_rules("icann"))
nrow(psl_rules("private"))
```

---

psl_use *Choose the active Public Suffix List for this session*

Description

Switches the list backing every query in the current R session. The change is session-only and is validated before any session state changes; a failure leaves the previously active list usable. A successful switch invalidates the match-result cache.

Usage

```
psl_use(source = c("bundled", "cache", "path"), path = NULL)
```

Arguments

source	Where to load the list from: "bundled" (the pinned package snapshot), "cache" (the latest successfully validated snapshot from psl_refresh()), or "path" (a custom file).
path	For source = "path", a single readable PSL-format UTF-8 file containing one complete ICANN section and one complete PRIVATE section, using official markers. Must be NULL for any other source.

Details

A custom path is held to the same runtime duplicate policy as `psl_refresh()`: exact same-section duplicates warn once and are deduplicated, while conflicting rule kinds for the same labels are fatal. Cache and custom-path sources are read in source form and indexed under the runtime normalizer; they never reuse the bundled generated index.

Value

Invisibly, the `psl_version()` row for the newly active list.

See Also

`psl_refresh()`, `psl_version()`, `psl_rules()`

Examples

```
psl_use("bundled")
## Not run:
psl_use("cache")
psl_use("path", path = "my_list.dat")

## End(Not run)
```

`psl_version` *Identity of the active Public Suffix List*

Description

Returns a one-row data.frame describing the list currently active in this R session: its source-snapshot provenance and the normalization identifiers actually used to index the active matcher. Reproducing a query result requires both the active-list identity and these normalization identifiers (PRD s10), so a reproducibility-sensitive workflow should record this row.

Usage

```
psl_version()
```

Details

The columns, in order, are:

`source` `"bundled"`, `"cache"`, or `"path"`.

`path` File path of a `"cache"` or `"path"` source; `NA` otherwise.

`retrieved_at` Network retrieval timestamp, or `NA`.

`list_date` Upstream list date, or `NA` when unknown.

`commit` Upstream commit SHA, or `NA` when unknown.

`size` Source byte size (integer).

`checksum` Source checksum, including its algorithm prefix (e.g. `"sha256:..."`).

`normalizer` The dependency providing canonicalization, currently `"punycoder"`.

`normalizer_version` Its installed package version.

`normalization_profile` Its stable case-mapping / IDNA / validation profile identifier.

`unicode_version` The Unicode data version used by that profile.

Unavailable metadata is a typed `NA`, never omitted. The normalization identifiers describe the implementation used by the current session, whether the active list came from the bundled snapshot, the user cache, or a custom path; an in-memory compatibility rebuild (PRD s8.3) updates them without altering the shipped source identity or checksum.

Value

A one-row base data.frame with the columns described in Details.

See Also

`psl_use()`, `psl_refresh()`, `psl_rules()`

Examples

```
psl_version()
```

`public_suffix` *Public suffix of a host*

Description

Returns the public suffix (effective top-level domain, eTLD) of each host under the selected Public Suffix List policy, following the official prevailing-rule algorithm.

Usage

```
public_suffix(
  domain,
  section = c("all", "icann", "private"),
  output = c("ascii", "unicode"),
  unknown = c("default", "na"),
  invalid = c("na", "error")
)
```

Arguments

domain	Character vector of DNS hostnames (not URLs). Each element may be a mixed-case ASCII, Unicode, or A-label hostname, a single label, or a hostname with exactly one terminal root dot. See **Input contract**.
section	Which rule sections are eligible: "all" (default; ICANN and PRIVATE), "icann", or "private". Section filtering happens before prevailing-rule selection, so "private" does not silently add ICANN rules; a host matching no rule in the section falls through to the implicit default rule unless unknown = "na".
output	"ascii" (default) returns lowercase A-labels; "unicode" decodes them after matching. A terminal root dot is preserved either way.
unknown	"default" (default) applies the spec's implicit * rule, so an unlisted single label is its own public suffix; "na" returns NA when no explicit rule in the selected section matches.
invalid	"na" (default) returns NA for each invalid element without a warning; "error" aborts on the first invalid element, reporting its 1-based index.

Value

A character vector with length(domain), preserving the names of domain. Other attributes are dropped.

Input contract

NA is treated as missing (returns NA), not invalid. Invalid elements include empty or whitespace-only strings, leading or consecutive dots, URL syntax, IPv6 addresses, canonical dotted-decimal IPv4 literals, and labels that fail hostname/IDNA validation. Wrong argument types and non-scalar or unknown option values always abort regardless of invalid.

See Also

registrable_domain(), is_public_suffix(), suffix_extract(), public_suffix_rule()

Examples

```
public_suffix("www.example.com")
public_suffix("example.co.uk")
public_suffix("example.com.")
public_suffix("madeuptld", unknown = "na")
```

---

public_suffix_rule *Inspect the prevailing PSL rule for each host*

Description

Inspect the prevailing PSL rule for each host

Usage

```
public_suffix_rule(
  domain,
  section = c("all", "icann", "private"),
  unknown = c("default", "na"),
  invalid = c("na", "error")
)
```

Arguments

domain	Character vector of DNS hostnames (not URLs). Each element may be a mixed-case ASCII, Unicode, or A-label hostname, a single label, or a hostname with exactly one terminal root dot. See **Input contract**.
section	Which rule sections are eligible: `"all"` (default; ICANN and PRIVATE), `"icann"`, or `"private"`. Section filtering happens before prevailing-rule selection, so `"private"` does not silently add ICANN rules; a host matching no rule in the section falls through to the implicit default rule unless `unknown = "na"`.
unknown	`"default"` (default) applies the spec's implicit `*` rule, so an unlisted single label is its own public suffix; `"na"` returns `NA` when no explicit rule in the selected section matches.
invalid	`"na"` (default) returns `NA` for each invalid element without a warning; `"error"` aborts on the first invalid element, reporting its 1-based index.

Value

A base data.frame with one row per input and columns, in order: `input` (original), `host_ascii` (canonical A-label host), `rule` (the canonical rule including `*.` or `!`, `"*"` for the implicit default), `kind` (`"normal"`, `"wildcard"`, `"exception"`, or `"default"`), `rule_section` (`"icann"`, `"private"`, or `NA` for the default/no result), and `public_suffix_ascii` (the derived A-label public suffix). Invalid rows are `NA` in every derived column. A valid host left unresolved by `unknown = "na"` keeps `host_ascii` while the rule and suffix columns are `NA`. An exception `rule` retains its `!` for auditability. Zero-length input returns a zero-row frame; all-invalid input keeps one row per input.

Input contract

`NA` is treated as missing (returns `NA`), not invalid. Invalid elements include empty or whitespace-only strings, leading or consecutive dots, URL syntax, IPv6 addresses, canonical dotted-decimal IPv4 literals, and labels that fail hostname/IDNA validation. Wrong argument types and non-scalar or unknown option values always abort regardless of `invalid`.

See Also

`public_suffix()`, `suffix_extract()`

Examples

```
public_suffix_rule("www.example.co.uk")
public_suffix_rule("madeuptld")
```

`registrable_domain` *Registrable domain of a host*

Description

Returns the registrable domain (eTLD+1) of each host: its public suffix plus one host label to the left. It is `NA` when no such label exists (the host is itself a public suffix) or when the public suffix is `NA`.

Usage

```
registrable_domain(
  domain,
  section = c("all", "icann", "private"),
  output = c("ascii", "unicode"),
  unknown = c("default", "na"),
  invalid = c("na", "error")
)
```

Arguments

`domain`	Character vector of DNS hostnames (not URLs). Each element may be a mixed-case ASCII, Unicode, or A-label hostname, a single label, or a hostname with exactly one terminal root dot. See **Input contract**.
`section`	Which rule sections are eligible: `"all"` (default; ICANN and PRIVATE), `"icann"`, or `"private"`. Section filtering happens before prevailing-rule selection, so `"private"` does not silently add ICANN rules; a host matching no rule in the section falls through to the implicit default rule unless `unknown = "na"`.
`output`	`"ascii"` (default) returns lowercase A-labels; `"unicode"` decodes them after matching. A terminal root dot is preserved either way.
`unknown`	`"default"` (default) applies the spec's implicit `*` rule, so an unlisted single label is its own public suffix; `"na"` returns `NA` when no explicit rule in the selected section matches.
`invalid`	`"na"` (default) returns `NA` for each invalid element without a warning; `"error"` aborts on the first invalid element, reporting its 1-based index.

Value

A character vector with `length(domain)`, preserving the names of `domain`. Other attributes are dropped.

Input contract

NA is treated as missing (returns NA), not invalid. Invalid elements include empty or whitespace-only strings, leading or consecutive dots, URL syntax, IPv6 addresses, canonical dotted-decimal IPv4 literals, and labels that fail hostname/IDNA validation. Wrong argument types and non-scalar or unknown option values always abort regardless of `invalid`.

See Also

`public_suffix()`, `is_public_suffix()`, `suffix_extract()`

Examples

```
registrable_domain("www.example.co.uk")
registrable_domain("com")
registrable_domain("foo.madeuptld", unknown = "na")
```

---

suffix_extract *Split hosts into subdomain, registrant label, and public suffix*

Description

Split hosts into subdomain, registrant label, and public suffix

Usage

```
suffix_extract(
  domain,
  section = c("all", "icann", "private"),
  output = c("ascii", "unicode"),
  unknown = c("default", "na"),
  invalid = c("na", "error")
)
```

Arguments

`domain`	Character vector of DNS hostnames (not URLs). Each element may be a mixed-case ASCII, Unicode, or A-label hostname, a single label, or a hostname with exactly one terminal root dot. See **Input contract**.
`section`	Which rule sections are eligible: `"all"` (default; ICANN and PRIVATE), `"icann"`, or `"private"`. Section filtering happens before prevailing-rule selection, so `"private"` does not silently add ICANN rules; a host matching no rule in the section falls through to the implicit default rule unless `unknown = "na"`.
`output`	`"ascii"` (default) returns lowercase A-labels; `"unicode"` decodes them after matching. A terminal root dot is preserved either way.
`unknown`	`"default"` (default) applies the spec's implicit `*` rule, so an unlisted single label is its own public suffix; `"na"` returns NA when no explicit rule in the selected section matches.

`invalid` "na" (default) returns NA for each invalid element without a warning; "error" aborts on the first invalid element, reporting its 1-based index.

Value

A base data.frame with one row per input and columns, in order: `input` (original, unchanged), `host` (canonical host in `output` form), `subdomain` (labels left of the registrable domain; "" when none), `domain` (the single registrant label left of the suffix), `suffix` (the public suffix), and `registrable_domain` (eTLD+1). `domain`, `subdomain`, and `registrable_domain` are NA when the host is itself a public suffix. If public-suffix resolution is NA, every derived column except `input` and a successfully normalized `host` is NA. Zero-length input returns a zero-row frame; all-invalid input keeps one row per input. Root dots are preserved on `host`, `suffix`, and `registrable_domain` only.

Input contract

NA is treated as missing (returns NA), not invalid. Invalid elements include empty or whitespace-only strings, leading or consecutive dots, URL syntax, IPv6 addresses, canonical dotted-decimal IPv4 literals, and labels that fail hostname/IDNA validation. Wrong argument types and non-scalar or unknown option values always abort regardless of `invalid`.

See Also

`public_suffix()`, `public_suffix_rule()`

Examples

```
suffix_extract("www.example.co.uk")
suffix_extract(c("example.com", "com", NA))
```

Index